Network Working Group   Steven M. Bellovin
Internet Draft   AT&T Labs Research

Expiration Date: October 2003   April 2003

Guidelines for Mandating Automated Key Management

draft-bellovin-mandate-keymgmt-00.txt

Status of this Memo

Abstract

The question often arises of whether or not a given security system requires some form of automated key management, or whether manual keying would suffice. This memo proposes guidelines for making such decisions; the presumption is that automated key management is generally but not always needed; if manual keying is proposed, the burden of proof is on the proposer.

1. Introduction

The question often arises of whther or not a given security system requires some form of automated key management, or whether manual keying would suffice.

There is no one answer to that question; circumstances differ. In general, automated key management SHOULD be used. Occasionally, relying on manual key management is reasonable; we propose some guidelines for making that judgment.

On the other hand, relying on manual key management has its disadvantages. We thus outline concerns that would suggest that manual key management would be a bad idea.

1.1. Terminology

The keywords MUST, MUST NOT, REQUIRED, SHALL, SHALL NOT, SHOULD, SHOULD NOT, RECOMMENDED, MAY, and OPTIONAL, when they appear in this document, are to be interpreted as described in [RFC2119].

2. Requirements

These are a set of guidelines, not rules, for evaluating when automated key management should or shouldn't be used. Informed judgment is necessary when applying them.

In this context, "key management" is automatic derivation of session key(s), as opposite to long-term keys used to authenticate the derived key(s). How this long-term key gets to the talking entities and what kind of a key it is (pre-shared secret, RSA public key, DSA, you-name-it) is beyond the scope of this document. Examples of key management systems include IKE and Kerberos; S/MIME and TLS include key management functions.

A session key is used to protect application data.

In general, automated key management SHOULD be used. This is a very strong "SHOULD".

Key management MUST be used if:

A central party will have to manage n^2 static keys.

A stream cipher such as RC4 or AES counter mode [AESMODE] is

used.

Long-lived session keys are used by more than two parties. (Except for multicast, this is a dubious situation in the first place, and should generally be discouraged no matter what.)

The likely operational environment is one where personnel (or device) turnover is reasonably frequent, thus creating a requirement for frequent rekeying.

Even manually-keyed systems need some provision for key changes; there must be some way to indicate which key is in use, to avoid problems during transition. Designs should sketch plausible mechanisms for deploying new keys and/or revoking compromised keys. If done well, such mechanisms can later be used by an add-on key management scheme.

Lack of automated key management can lead to vulnerabilities, including (but not limited to) cryptographic weaknesses or loss of some functionality, such as replay protection.

Key management software is not always large or bloated; even IKEv1 can be done in <200K, and TLS in half that much space. (TLS includes other functionality as well.)

Lack of clarity about who the principals are is not a valid reason for avoiding key management. Rather, it tends to indicate a deeper problem with the underlying security model.

Key management schemes should not be designed by amateurs; it is almost certainly inappropriate for WGs to design their own. To put it in concrete terms, the very first key management protocol in the open literature was published in 1978. A flaw was published in 1983. The fix proposed in 1983 was cracked in 1994. In 1996, a new flaw was found in the original 1978 version, in an area not affected by the 1983/1994 issue. All of these flaws were blindingly obvious once described -- but no one spotted them earlier. Note that the original protocol (translated to know about certificates, which hadn't been invented at the time) was only 3 messages.

Situations where a desire to avoid key mangement may be reasonable include:

> Very limited available bandwidth or very high round-trip times. There are interactions here -- public key systems tend to require long messages and lots of computation; symmetric key alternatives, such as Kerberos, often require several round trips and interaction with third parties.

> Low value of the information

> Very limited scale of each deployment

Note that assertions about such things should often be viewed with the skepticism afforded to claims that "this will only be used on a LAN or two". In other words, the burden of demonstrating that manual key management is appropriate should be on the proponents --- and it's a fairly high barrier that they need to overcome.

3. References

[AESMODE] "Recommendation for Block Cipher Modes of Operation - Methods and Techniques", NIST Special Publication SP 800-38A, December 2001.

[RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner. RFC 2119. March 1997.

4. Author Information

Steven M. Bellovin
AT&T Labs Research

Shannon Laboratory
180 Park Avenue
Florham Park, NJ 07932
Phone: +1 973-360-8656
email: bellovin@acm.org